No New Matter:

Support for the new claims is found in the originally filed disclosure. No new matter is presented.

In particular, as in new Claim 32, "*generating a mask that indicates relevant bits that are relevant to identify the rule and do not care bits that are extraneous for identifying the rule*" is disclosed at paragraph 39, Table 1; „*generating a vector comprised of memory positions and at least a corresponding bit for each of the memory positions, by setting bits for those memory positions of the vector pointed to by bits of the rule when read as an address, the address including both relevant bits and do not care bits, to a value that indicates that the rule is detected*" is disclosed paragraphs 39 – 41 and illustrated by Tables 1 and 2; „*inserting values of the bits of the vector into memory positions of a memory corresponding to the memory positions for each bit of the vector*" is disclosed at paragraph 43; „*dividing the bits of the data key into a plurality of chunks*„ is disclosed at paragraph 28 and throughout; „*extracting data from the memory at an address corresponding to the value of each of at least some of the chunks;* „ is disclosed at paragraph 32 and throughout; „*examining the data extracted from each memory address corresponding to the at least some of the chunks to determine if the rule is obeyed for the entire data key.* „ is disclosed at paragraphs 33-36 and throughout.

In Claim 33 the "*step of generating a vector comprises automatically setting the value of the corresponding bits of the vector to indicate that the rule is detected according to the following function: Rule[I][C bits] AND Mask[I][C bits] is equal to Mask[C bits] AND J,wherein I is the Ith rule, C is the number of bits in a chunk, and J ranges from 0 to* 2^C *-1*" is disclosed at paragraph 44.

In Claim 34 the "*step of generating a mask generates a string of bits corresponding to the number of bits in a rule, and sets each bit in the mask that corresponds to a bit in the rule that is relevant for identifying that rule*" is disclosed at paragraph 39 and Table 1.

In Claim 35 the "*step of generating a vector comprises setting the value of bits corresponding to a number of bits comprising a chunk*" is disclosed paragraph 43.

In Claim 36 the "*step of configuring the memory into a two dimensional memory structure, a first dimension corresponding to the chunks of the data key and a second dimension corresponding to different rules*" is disclosed at paragraph 15 and shown diagrammatically in Fig. 1.

In Claim 37 "*examining the data further comprises performing at least one AND operation on the data extracted from the memory corresponding to each of the chunks of the data key to determine if the rule is obeyed for the entire data key*„ is disclosed at for example paragraphs 35-36.

In Claim 38 the "*step of providing the memory as separate memory devices and allocating sub sets of the different memory devices based on the number of rules to be detected*„ is disclosed at paragraph 32 and diagrammatically illustrated in Fig. 1.

In Claim 39 the recitation "*wherein a number of rules are detected, further comprising the step of prioritizing the rules according to a predetermined order of priority for rules*„ is disclosed in paragraph 37.

In Claim 40, the "*step of parsing for parsing a packet received over a network into a data key*„ is disclosed at paragraph 5.

In Claim 41, the *"step of receiving the packet from an Internet based network*„ is disclosed at paragraph 3.

In Claim 42 *"A system for comparing a data key to a rule, wherein the data key and rule are representable by bits, the system including: a memory comprised of memory positions and at least a corresponding bit for each of the memory positions making up a vector, wherein bits for those memory positions of the vector pointed to by bits of the rule when read as an address, the address including both relevant bits and do not care bits, are set to a value that indicates that the rule is detected"„an interface that divides the bits of the data key into a plurality of chunks; „„a comparator that compares data from each memory address corresponding to the at least some of the chunks to determine if the rule is obeyed for the entire data key*„ is disclosed at paragraphs 33-36 and throughout.

In Claim 43 the „ *setting the value of the corresponding bits of the vector stored in memory to indicate that the rule is detected according to the following function:"Rule[I][C bits] AND Mask[I][C bits] is equal to Mask[C bits] AND J,"wherein I is the Ith rule, C is the number of bits in a chunk, and J ranges from 0 to* 2^C *-1,"and wherein Mask are bits set to a value indicating relevant bits that are relevant to identify the rule and do not care bits that are extraneous for identifying the rule*„ is disclosed at paragraph 44.

In Claim 44 "*a mask formed of a string of bits corresponding to the number of bits in a rule, and sets each bit in the mask that corresponds to a bit in the rule that is relevant for identifying that rule"* is disclosed at paragraph 39 and Table 1.

In Claim 45 *"the memory stores the value of bits of the vector corresponding to a number of bits comprising a chunk"* is disclosed paragraph 43.

In Claim 46 "*the memory is configured into a two dimensional memory structure, a first dimension corresponding to the chunks of the data key and a second dimension corresponding to different rules"* is disclosed at paragraph 15 and shown diagrammatically in Fig. 1.

In Claim 47 *"the comparator includes at least one AND device that operates on data extracted from the memory corresponding to each of the chunks of the data key to determine if the rule is obeyed for the entire data key*„ is disclosed at for example paragraphs 35-36.

In Claim 48 *"the memory is provided as separate memory devices and further comprising a switch that allocates the which memory devices are grouped together based on the number of rules to be detected*„ is disclosed at paragraph 32 and diagrammatically illustrated in Fig. 1.

In Claim 49 *"a prioritzer that prioritizes rules according to a predetermined order of priority for rules*„ is disclosed in paragraph 37.

In Claim 50 "*a parser for parsing a packet received over a network into a data key"* is disclosed at paragraph 5.

In Claim 51 *"receiving the packet from an Internet based network,,* is disclosed at paragraph 3.

In Claim 52 *The system of claim 42, wherein the memory includes only a number of individual memory units (per bit) according to the formula: NM/(2^C L^2 C), wherein, N is the number of bits in the data key, M is the number of rules available for detection, C is the number of bits in a chunk and L is the width of a discrete memory device,,* is disclosed at paragraph 47.

Substantive Response:

New Claims 32-52 are now present in this application, of which Claims 32 and 42 are Independant.

These claims are presented to overcome Lu US Patent 6,778,984, which was the only reference cited in the previous Final Rejection of the claims.

In particular, the new claims were carefully constructed to capture the invention as described in the patent application as originally filed, to wit, Independent Claim 32 is a method claim that recites *„generating a vector comprised of memory positions and at least a corresponding bit for each of the memory positions, by setting bits for those memory positions of the vector pointed to by bits of the rule when read as an address, the address including both relevant bits and do not care bits, to a value that indicates that the rule is detected"*. Independent Claim 42 is directed to a system and includes a similar feature.

By contrast, Lu '984 does not store a vector that includes *"values of bits that indicate the rule is detected."* Rather, Lu simply stores the rules themselves in memory. He discloses, for example, *"various combinations of the bits inside each sub-key value and same sub-key value field to of each rule are compared [sic]"* at Col. 4, lines 24-26.

The difference between the newly added Claims 32 and 42 and Lu is significant. In the present invention, because the values of the result are stored as a vector (for example, the bit value 1), there is no comparison that needs to be done by a processor. In the present invention, only the bit value of "1" needs to be read out and the processor knows that the rule is being presented in the data key. Hence, no logical operations need to be performed with the present invention to detect the rule. Indeed, the only logical operations needed is performed by AND gates that "add up" the bit values read out from a number of memory positions corresponding to the chunks of the data key. *Please see paragraph 14 in the instant description.*

Lu, on the other hand, is focused on search algorithms and simplifying the searching. For this he has concocted an elaborate set of algorithms for populating his memory. *Please see, for example, Lu at Col. 6, lines 54-60.*

For this reason alone Lu should be withdrawn as a reference and Claims 32-52 should now be indicated as allowable.

In addition, the present invention as claimed in Claims 32 and 42*"sets bits for those memory positions of the vector pointed to by bits of the rule when read as an address, the address including both relevant bits and do not care bits, to a value that indicates that the rule is detected"*. In other words, the present invention as claimed sets the values of the

memory at the locations corresponding to the address of the rule for all of those addresses indicated by combinations of the relevant bits and the possible do not care bits.

Lu simply describes no such recitation. As described above, Lu does not even describe storing a vector of values indicating detection of a rule. It is not surprising, therefore, that Lu does not disclose storing bit values in a number of address locations (ie at those locations indicated by the combination of relevant bits and do not care bits).

It is clear that the present invention as claimed and Lu are directed to vastly different ways to populate the memory. The algorithm for generating the vector and, thus, populating the memory, in the present invention is given by the formula: Rule[I][C bits] AND Mask[I][C bits] is equal to Mask[C bits] AND J. While the essence of the invention is already spelled out in the independent claims, dependent claims 33 and 43 specifically recite this formula.

Lu nowhere comes close to populating his memory in the particular manner now claimed in Claims 33 and 43.

Another advantage of the present invention as claimed is that it optimizes the use and amount of memory needed to detect rules. Indeed, with the present invention, only $NM/(2^C L^2 C)$ bits of memory are needed to detect rules. This formula is now positively recited in Claim 52.

By contrast, Lu specifically identifies the number of memory devices he needs (per bit) at Col. 4 line 66 as $S = (W/G)xNx2^G$, where S is memory size (in bits), W is Width of the rule, G is the number of bits in his sub field and N is the number of rules. By simply replacing the terminology used by Lu with that used in the instant disclosure, we obtain Lu's formula in terms of the present invention as follows: $(N/C)xMx2^C$.

Simply comparing these two formulas:

Present Invention (Memory Needed)	Lu's Formula (Memory Needed)
$NM/(2^C L^2 C)$	$(N/C)xMx2^C$ or, $NM2^C/C$

we see instantly that Lu's device requires vastly more memory devices (per bit), actually a factor of $2^C 2^C L^2$,or $2^{2C} L^2$ MORE devices required by Lu than the present invention.

Clearly, the present invention as claimed in Claims 32-52 is patentably distinct from Lu '984.

Request for reconsideration and allowance of Claims 32-52 are respectfully requested.

Respectfully submitted,

/Ignacio Marc Asperas/

Ignacio Marc Asperas, US Reg. No. 37,274
Lantiq Beteiligungs Gmbh, & Co. k.g.
Legal Dept. & representative of Lantiq Deutschland
10 Feb. 2010